C1 cont'd

decoding each bit-plane to provide data to the target device, said step of decoding comprising the step of truncating each bit-plane in the embedded codestream for data necessary to support the target device.

a1

69. (New) The method defined in Claim 68 wherein each bit-plane is truncated based on an indication in each coding unit denoting a location where truncation may occur.

Sub E1

70. (New) The method defined in Claim 69 wherein the indication comprises a marker.

71. (New) The method defined in Claim 69 wherein the indication comprises a pointer.

72. (New) The method defined in Claim 68 wherein each bit-plane is truncated based on one of a plurality of indications in each coding unit denoting locations where truncation may occur, said step of truncating further comprising the step of selecting one of the indications based on the target device.

Sub C2

73. (New) The method defined in Claim 68 wherein the step of truncating comprises truncating target resolution coefficients, coded separately in each coding unit, from the embedded codestream.

74. (New) The method defined in Claim 68 wherein the target device comprises a low resolution, high pixel depth embedded target such that the step of decoding decodes as many higher level coefficients as needed to achieve full pixel depth and low spatial resolution of the target device.

75. (New) The method defined in Claim 68 wherein the target device comprises a high resolution, low pixel depth embedded target and truncating each coding unit at a number of bit-planes and inverse wavelet transforming the non-truncated data of each coding unit to achieve the low pixel depth and high spatial resolution of the target device.

76. (New) The method defined in Claim 68 further comprising the steps of:

selecting coding units based on an amount of available buffering at the target device; and

truncating each coding unit with more data than available buffering.

77. (New) The method defined in Claim 68 wherein the step of truncating further comprises the steps of:

determining a uniform amount to truncate each coding unit;

truncating at least a portion of at least one importance level in each coding unit.

78. (New) The method defined in Claim 77 wherein the step of truncation is performed using information in a header of the codestream setting forth importance level information.

79. (New) The method defined in Claim 77 wherein the step of truncation is performed using information in a header of the codestream setting forth importance level information for each coding unit in the codestream.

80. (New) The method defined in Claim 77 wherein the steps are performed after encode time.

81. (New) An encoder for encoding input data in a compressed data stream for transmission on a channel, said entropy coder comprising:

a reversible wavelet filter for transforming the input data into a plurality of coefficients;

an embedded coder coupled to the reversible wavelet filter for performing embedded coding on the plurality of coefficients, such that a bit stream is generated;

an entropy coder coupled to the embedded coder for performing entropy coding on the bit stream to create coded data,

a channel manager for controlling the storage of compressed data in a fixed size memory, wherein the memory manager divides compressed data into importance levels, and further wherein the channel manager transmits all data sufficient bandwidth is available, otherwise less or important data is discarded and only more important data is transmitted.

82. (New) The encoder defined in Claim 81 wherein the channel manager dynamically determines the bandwidth of the channel.

83. (New) The encoder defined in Claim 81 wherein the data is divided up into tiles, and each tile is separately coded and divided by importance level into fixed size pages.

84. (New) The encoder defined in Claim 81 wherein the channel manager further comprising a buffer memory.

85. (New) The encoder defined in Claim 84 wherein the buffer memory is circular buffer memory having multiple fixed size segments, such that the data is output as each fixed size segment is completely allocated.

86. (New) The encoder defined in Claim 85 wherein the packet size matches the bandwidth of the channel.

87. (New) An encoder for encoding input data in a compressed data stream for transmission on a channel, said entropy coder comprising:

a reversible wavelet filter for transforming the input data into a plurality of coefficients;

an embedded coder coupled to the reversible wavelet filter for performing embedded coding on the plurality of coefficients, such that a bit stream is generated;

an entropy coder coupled to the embedded coder for performing entropy coding on the bit stream to create coded data; and

a channel manager coupled to the entropy coder to divide the data into fixed size segments, wherein each fixed size segment includes an indication of the most important data in the segment.

88. (New) The encoder defined in Claim 87 wherein each segment includes coded data, an optional pointer and the level of the most important data in this segment.

89. (New) The encoder defined in Claim 87 wherein the starting point of each band is identified using restart markers.